#2
2-13-03
ae

# 中華民國經濟部智慧財產局

INTELLECTUAL PROPERTY OFFICE
MINISTRY OF ECONOMIC AFFAIRS
REPUBLIC OF CHINA



茲證明所附文件，係本局存檔中原申請案的副本，正確無訛，
其申請資料如下 ：

This is to certify that annexed is a true copy from the records of this office of the application as originally filed which is identified hereunder:

申　　請　　日：西元 2001 年 04 月 03 日
Application Date

申　請　案　號：090218189
Application No.

申　　請　　人：國立交通大學
Applicant(s)
(西元 2001 年 8 月 27 日陳寶龍、李鎮宜先生將本案之專利申請權讓與國立交通大學)

局　　長
Director General

陳明邦

發文日期：西元 2001 年 12 月 04 日
Issue Date

發文字號：09011018709
Serial No.

| 申請日期： | 案號： |
|---|---|
| 類別： | |

（以上各欄由本局填註）

# 新型專利說明書

| | | |
|---|---|---|
| 一、新型名稱 | 中文 | 微處理器指令讀取法及其構造 |
| | 英文 | |
| 二、創作人 | 姓名（中文） | 1. 陳寶龍<br>2. 李鎮宜 |
| | 姓名（英文） | 1.<br>2. |
| | 國籍 | 1. 中華民國 2. 中華民國 |
| | 住、居所 | 1. 新竹縣新豐鄉青埔村2鄰32號<br>2. 新竹市博愛街75之1號 |
| 三、申請人 | 姓名（名稱）（中文） | 1. 國立交通大學 |
| | 姓名（名稱）（英文） | 1. |
| | 國籍 | 1. 中華民國 |
| | 住、居所（事務所） | 1. 新竹市大學路1001號 |
| | 代表人姓名（中文） | 1. 張俊彥 |
| | 代表人姓名（英文） | 1. |

四、中文創作摘要　（創作之名稱：微處理器指令讀取法及其構造）

一種微處理器指令讀取法及其構造，於指令執行程序中，當預解讀取下一執行指令為條件式跳躍指令時，於下一指令週期，程式記憶體才需讀取二個指令，其他僅預讀取下一指令，其係利用處理單元對解譯之下一執行指令來設定指令讀取數目暫存器之狀態，該解譯之下一執行指令若為條件式跳躍指令，則該指令讀取數目暫存器會同時致能奇、偶數緩衝暫存器，程式記憶體以讀取二個指令，而其他所解譯之下一執行指令，該指令讀取數目暫存器僅會致能奇、偶數緩衝暫存器其一者，程式記憶體只讀取一指令，以達程式記憶體不必要的讀取，進而降低電量消耗。

英文創作摘要　（創作之名稱：）

本案已向

| 國(地區)申請專利 | 申請日期 | 案號 | 主張優先權 |
| --- | --- | --- | --- |
| - | 無 | | |

五、創作說明 (1)

創作領域

本創作係關於一種微處理器指令讀取法及其構造，特別指一種預解讀取指令為條件式跳躍指令時，程式記憶體需讀取二指令，其它情形，均只需讀取之一指令，達成不必要的程式記憶體讀取，以降低電量消耗者。

創作背景

以單一電腦對使用者來看，其處理程式指令的時間可反應出其效能所在，是故，如何降低微處理器執行程式指令的反應時間係為重要的研究課題。單一週期指令微處理器係藉由一指令週期中來完成一指令之執行，並讀取下一預執行之指令，藉此於每一指令週期完成一指令之執行，然實際上於指令群中，非所有指令均數單週期完成者，請參閱第一圖所示，於指令週期中執行之指令為如邏輯運算之常態指令，藉由PC(程式計數器)+1連續性讀取下一預執行之指令，則可於每一指令週期完成一指令之執行，若下一預執行之指令為呼叫指令(CALL)時，因PC不是按照一般加一方式之連續性讀取下一指令方式處理，而造成PC發生不連續，於此，需於其後插入一無執行指令(NOP)，藉由一指令週期執行NOP來載入CALL指令所呼叫之正確位址"M"於PC，而需耗費額外指令週期方能執行呼叫之指令，而影響微處理器執行程式之效能，於此，現行有一種微處理器指令讀取法，利用一種預先讀取指令及解譯方式，以改善前述耗時之問題，其方法請參閱第二圖所示，於指令執行

程序中，執行第N指令時，並預讀取位址為N+1、N+2之指令，且同時對第N+1指令解譯，而於第N+1指令做解譯發現非如邏輯運算之常態指令(如CALL指令)時，則於下一指令週期，執行之指令以NOP取代，而載入正確位址"M"，並預讀取位址為M、M+1之指令，而於下一指令週期，即可執行所呼叫位址M之指令，而可改善前述需額外指令週期方能完成，而得提昇處理效能，然此於每一指令週期利用預讀取二指令儲存於程式記憶體，卻會造成電量耗費之問題，若能進一步改善其電量耗費之問題，則具正面意義。

創作目的

本創作之目的係在提供一種微處理器指令讀取法，於執行指令程序中，解譯之下一執行指令只有當為條件式跳躍指令時，才須預讀取下兩個指令，其他僅預讀取下一指令，以達成不必要的程式記憶體讀取，進而降低電量消耗。

本創作另一目的係在提供一種微處理器指令讀取構造，於執行指令程序中，利用處理單元對解譯之下一執行指令來設定指令讀取數目暫存器之狀態，該解譯之下一執行指令若為條件式跳躍指令，則該指令讀取數目暫存器會同時致能奇、偶數緩衝暫存器，程式記憶體以讀取二個指令，而其它所解譯之下一執行指令，該指令讀取數目暫存器僅會致能奇、偶數緩衝暫存器其一者，程式記憶體以讀取一指令，以達程式記憶體不必要的讀取，進而降低電量消

耗 。

創作詳細說明

一般而言，指令可分四大類，第一類為如執行邏輯運算之常態指令，第二類為無條件跳躍指令，第三類為呼叫和返回指令，第四類為條件式跳躍指令，而第一類常態指令，其下一讀取指令的位址為PC+1，位址只有一個，第二類無條件跳躍指令，雖然為新位址，但位於其指令上，因此位址只有一個，第三類呼叫和返回指令，其呼叫的位址也在於其指令上，而返回指令之位址在堆疊上，因此位址只有一個，至於地四類條件式跳躍指令，其位址有二個可能，其一位址為PC+1，另一位址為PC+2，其位址則由處理單元所決定，是故，解譯之下一執行指令若為第四類條件式跳躍指令時，則特別須預讀取下二指令，以於下一指令週期執行時，可選取正確位址之指令執行。

請參閱第三圖所示，本創作微處理指令讀取法，於執行指令程序中，於執行指令步驟301後經預解讀取步驟302即進入四種流程，若解譯之下一執行指令為常態指令303，即令PC加一307，而指令讀取數目暫存器設為程式記憶體讀取一指令311之狀態；若解譯之下一執行指令為條件跳躍指令304，則令PC指向新位址308，而指令讀取數目暫存器設為程式記憶體讀取一指令311；若為呼叫或返回指令305，則令PC指向新位址309，而指令讀取數目暫存器

設為程式記憶體讀取一指令311；若為條件式跳躍指令306，則令PC指向下兩位址之指令310，而指令讀取數目暫存器設為程式記憶體讀取二指令312，而由處理單元決定選取其一之正確指令313來執行。

請參閱第四圖所示，為本創作微處理器指令讀取構造，本創作利用指令讀取數目暫存器411的設置，而於處理單元410預解讀取指令為條件式跳躍指令時，將該指令讀取數目暫存器411設為讀取二指令狀態（設以二進位數碼"1"表示），即於下一指令週期讀取二個指令，該處理單元410預解讀取指令為其它類指令者，則該指令讀取數目暫存器411設為"0"，而於下一指令週期讀取一個指令，儲存程式指令之程式記憶體構造，本創作程式記憶體區分奇、偶數頁程式記憶體407、406，當奇、偶數位址緩衝暫存器405、404被致能時，奇、偶數頁程式記憶體407、406才會被指令緩衝暫存器409讀取，茲配合例子說明如后：

【當指令讀取數目暫存器為"0"，位址線為"10"時】

加一電路401即產生位址"11"，而多工器402之選擇開關"S"連接位址線之最低位元，是故，多工器402選擇位址"10"，而多工器403之選擇開關"S"連接至加一電路401後之位址線之最低位元，而選擇位址"11"，而偶數位址緩衝暫存器404經由多工器414被致能，而會被偶數頁程式記憶體406讀取，而指令緩衝暫存器409透過多工器408只會讀取其位址。

【當指令讀取數目暫存器為"0"，位址線為"11"時】

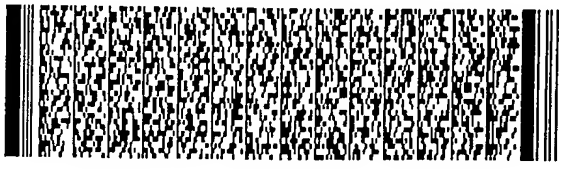
五、創作說明 (5)

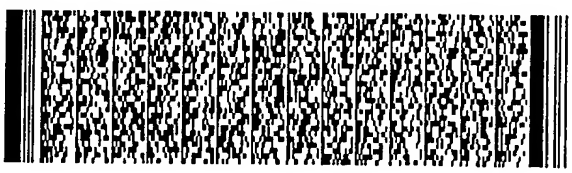
加一電路即產生位址"12"，多工器402選擇位址"12"，而多工器403選擇位址"11"，而奇數位址緩衝暫存器405經由多工器413被致能，而會被奇數頁程式記憶體407讀取，指令緩衝暫存器409透過多工器408只會讀取其位址。

【當指令讀取數目暫存器為"1"，位址線為"11"時】

加一電路即產生位址"12"，多工器402選擇位址"12"，多工器403而選擇位址"11"，而奇、偶數位址緩衝暫存器405、404均被致能，而位址"11"、"12"均會被奇、偶數頁程式記憶體407、406讀取，指令緩衝暫存器409所選取位址則由處理單元410所決定（因多工器412之選擇開關"S"為1，其輸出由"1"端進入，而該"1"端又為處理單元410所決定）。

茲再配合第五圖程式例說明本創作運作方式，該程式例說明處理位址"10"之條件式跳躍指令時，選取位址"11"之無條件跳躍指令，同時說明處理無條件跳躍指令時，讀取下一位址指令者，請參時序圖，於PC位址"10"之指令週期，執行指令位址為"9"之指令，程式記憶體所讀取之下一執行指令之位址為"10"，經解譯為條件式跳躍指令時，指令讀取數目暫存器設為"1"，而於下一指令週期，執行位址為"10"之指令，程式記憶體讀取位址"11"、"12"二指令，而處理單元選取位址"11"為下一執行之指令，經解譯為無條件跳躍指令時，指令讀取數目暫存器設為"0"，而於下一指令週期，執行位址"11"之指令以NOP取代，程式





記憶體讀取新位址"100"之指令，經解譯為常態指令，指令讀取數目暫存器設為"0"，而於下一指令週期，即可執行位址"100"之指令，而程式記憶體即讀取位址"101"之指令並將之解譯。

第六圖程式例與第五圖相同，其不同處在於條件式跳躍指令選取位址"12"之呼叫和返回指令，於PC位址為"200"之指令週期，執行位址"11"之指令以NOP取代，程式記憶體讀取新位址"200"之指令，經解譯為常態指令，指令讀取數目暫存器設為"0"，而於下一指令週期，即可執行位址"200"之指令，而程式記憶體讀取讀取位址"201"之指令，並解譯為返迴指令，而於下一指令週期，執行位址"201"之指令以NOP取代，程式記憶體讀取讀取返回位址"13"之指令並將之解譯，如是，本創作除預解讀取指令為條件式跳躍指令時，才需讀取二指令，其它情形(佔八成以上)，均只需讀取一指令，達成不必要的程式記憶體讀取，以降低電量消耗。

由上所述，本創作之微處理器指令讀取法及其構造實具新穎創作性與方便實用性，自已符合新型專利要件，懇請 鈞局貴審查委員能加以詳審，並賜准專利，以優惠民生，實感得便。

惟，以上所揭露之圖式及說明，僅為本創作之較佳實施例而已，非為用以限定本創作之實施，大凡熟悉該項技藝之人士其所依本創作之精神，所作之變化或修飾，皆應涵蓋在以下本案之申請專利範圍內。

圖式說明

第一圖為傳統微處理器指令讀取法之時序示意圖。

第二圖為另一傳統微處理器指令讀取法之時序示意圖。

第三圖為本創作微處理器指令讀取法之流程圖。

第四圖為本創作微處理器指令讀取構造之實施例圖。

第五圖、第六圖為本創作微處理器指令讀取法之時序示意圖。

圖號說明

401.加一電路　　402.多工器

403.多工器　　404.偶數位址緩衝暫存器

405.奇數位址緩衝暫存器　　406.偶數頁程式記憶體

407.奇數頁程式記憶體　　408.多工器

409.指令緩衝暫存器　　410.處理單元

411.指令讀取數目暫存器　　412.多工器

413.多工器　　414.多工器

六、申請專利範圍

1.一種微處理器指令讀取構造，其至少包含有：

處理單元，以執行算數邏輯、控制及預先讀取解譯位於指令緩衝暫存器之下一執行指令之運算；

指令緩衝暫存器，以讀取程式記憶體之指令位址；

程式記憶體，以儲存指令位址；及

指令讀取數目暫存器，以記錄指令讀取數目之狀態；

其係於指令執行程序中，利用處理單元預先讀取解譯下一執行指令來設定指令讀取數目暫存器之狀態，其中，

解譯之下一執行指令若為條件式跳躍指令，則設定指令讀取數目暫存器為讀取二指令狀態，而於下一指令週期，該程式記憶體讀取儲存二指令位置，而依該處理單元解譯而決定該指令緩衝暫存器讀取該程式記憶體中其一指令；

解譯之下一執行指令若為其他指令，則設定指令讀取數目暫存器為讀取一指令狀態，而於下一指令週期，該程式記憶體讀取儲存一指令位置，以達程式記憶體不必要的讀取而降低電量消耗者。

2.如申請專利範圍第1項所述之微處理器指令讀取構造，其中該些其它指令，若為算數邏輯之常態指令，於該下一指令週期該程式記憶體讀取下一指令(PC+1)位址，若為無條件跳躍指令，於該下一指令週期該程式記憶體讀取其指令指向之新位址，若為呼叫或返回指令，於該下一指令週期該程式記憶體讀取其指令指向之新位址或堆疊上之位址。

3.如申請專利範圍第1項所述之微處理器指令讀取構造，

六、申請專利範圍

其中該指令讀取數目暫存器係以二進位數碼"1"表示為讀取二指令之狀態，該指令讀取數目暫存器係以二進位數碼"0"表示為讀取一指令之狀態。

4.一種微處理器指令讀取構造，其包含：

指令讀取數目暫存器，以記錄指令讀取數目之狀態；

一處理單元，以執行算數邏輯、控制及預先讀取解譯位於指令緩衝暫存器之下一執行指令之運算；

指令緩衝暫存器，以讀取程式記憶體之指令位址；

程式記憶體，其包含奇、偶數頁程式記憶體，以儲存奇、偶數之指令位址；

位址緩衝暫存器，其包含奇、偶數位址緩衝暫存器，以存第一、第二多工器所選取奇、偶數之指令位址；

加一電路，以將位址線之指令位址加一；

第一多工器，以選取奇數指令位址；

第二多工器，以選取偶數指令位址；

第三多工器，以選取奇、偶數頁程式記憶體中之其一指令位址；

第四多工器，以致能偶數位址緩衝暫存器，而得讓偶數頁程式記憶體讀取其指令位址；

第五多工器，以致能奇數位址緩衝暫存器，而得讓奇數頁程式記憶體讀取其指令位址；

第六多工器，以控制第三多工器選取奇、偶數頁程式記憶體何者之指令位址；

於指令執行程序中，該處理單元預先讀取解譯之下一指

令來設定指令讀取數目暫存器之狀態，其中，

解譯之下一執行指令若為條件式跳躍指令，則設定指令數目暫存器為讀取二指令狀態，而於下一指令週期，該奇、偶數程式記憶體讀取儲存於奇、偶數位址緩衝暫存器之下兩個指令位址，而依該處理單元解譯而藉由第六多工器決定該指令緩衝暫存器讀取該程式記憶體中其一指令；

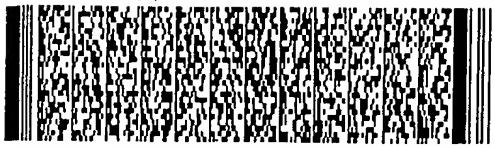
解譯之下一執行指令若為其它指令，則設定指令數目暫存器為讀取一指令狀態，而於下一指令週期，該程式記憶體讀取奇、偶數位址緩衝暫存器其一指令位址，以達程式記憶體不必要的讀取而降低電量消耗者。

5.如申請專利範圍第4項所述之微處理器指令讀取構造，其中該些其它指令，若為算數邏輯之常態指令，於該下一指令週期該程式記憶體讀取下一指令(PC+1)位址，若為無條件跳躍指令，於該下一指令週期該程式記憶體讀取其指令指向之新位址，若為呼叫或返回指令，於該下一指令週期該程式記憶體讀取其指令指向之新位址或堆疊上之位址。

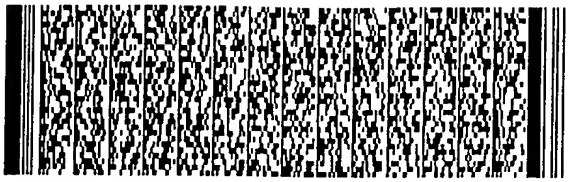
6.如申請專利範圍第4項所述之微處理器指令讀取構造，其中該指令讀取數目暫存器係以二進位數碼"1"表示為讀取二指令之狀態，該指令讀取數目暫存器係以二進位數碼"0"表示為讀取一指令之狀態。

申請案件名　微處理器指令讀取法及其構造

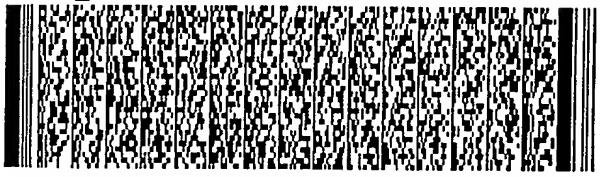
第 1/13 頁



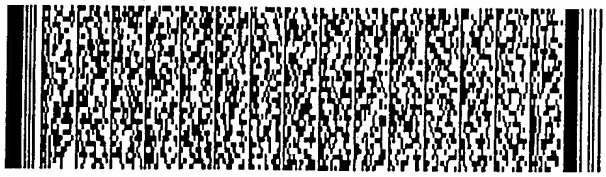
第 2/13 頁



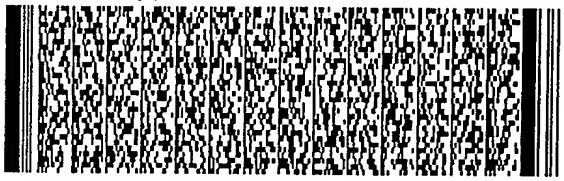
第 4/13 頁



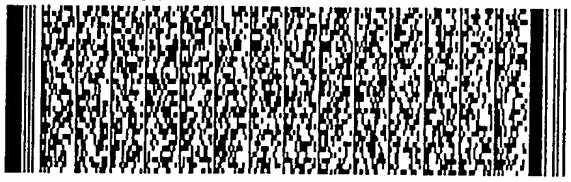
第 4/13 頁



第 5/13 頁

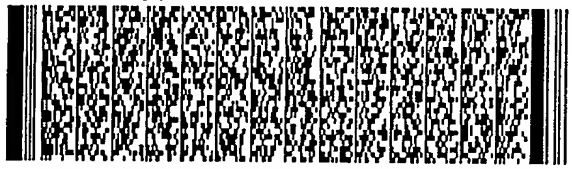


第 5/13 頁

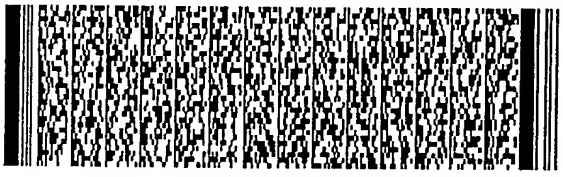


第 6/13 頁

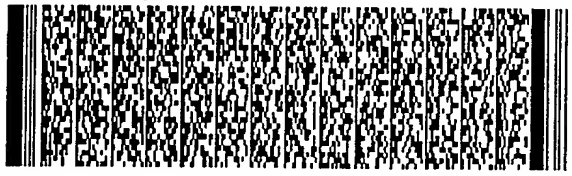
第 6/13 頁



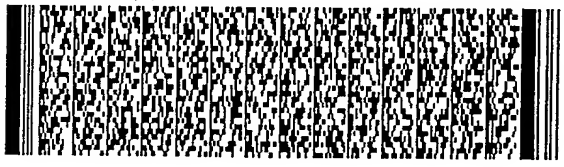
第 7/13 頁



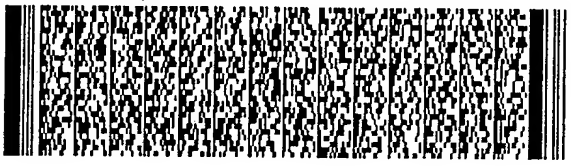
第 7/13 頁



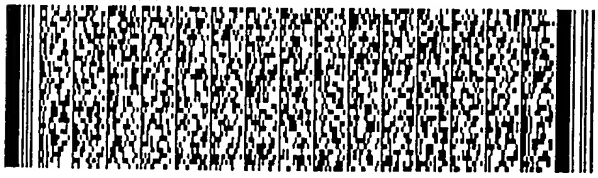
第 8/13 頁



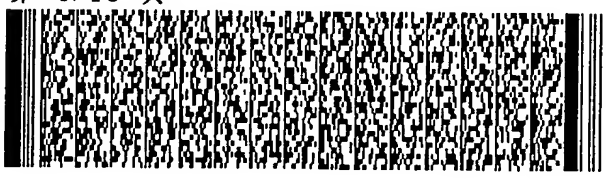
第 8/13 頁



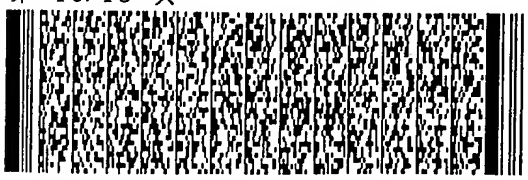
第 9/13 頁



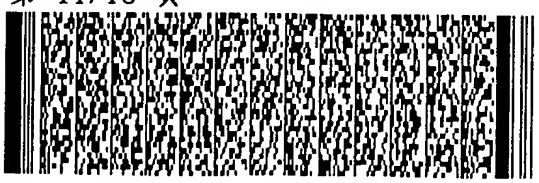
第 9/13 頁



第 10/13 頁

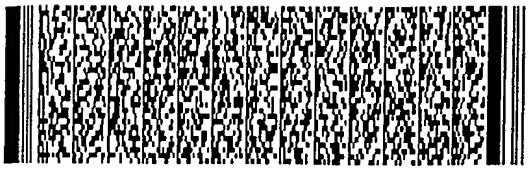


第 11/13 頁

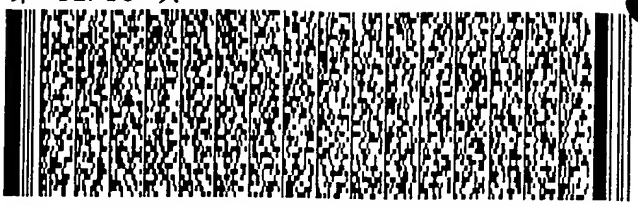


申請案件名 微處理器指令讀取法及其構造

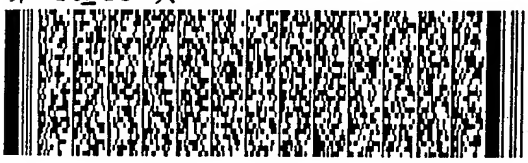
第 11/13 頁



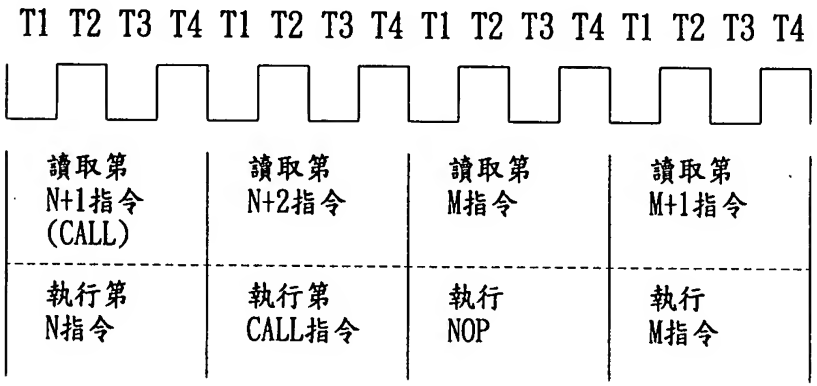
第 12/13 頁



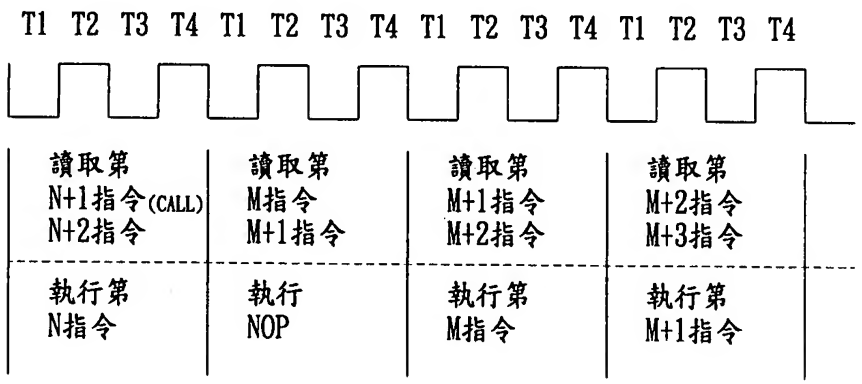
第 13/13 頁



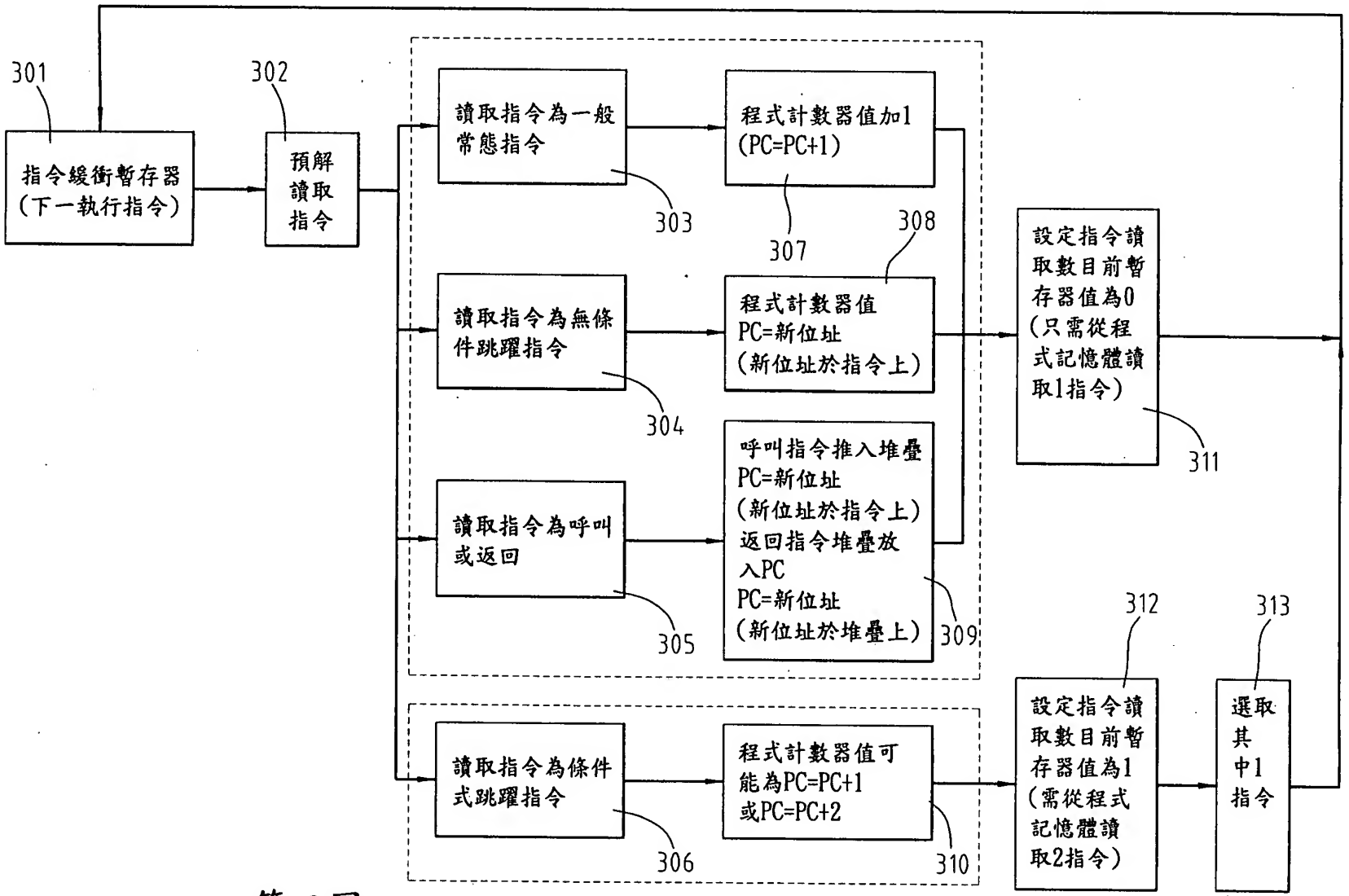
第 13/13 頁


T1 T2 T3 T4 T1 T2 T3 T4 T1 T2 T3 T4 T1 T2 T3 T4
讀取第 N+1指令 (CALL)
讀取第 N+2指令
讀取第 M指令
讀取第 M+1指令
執行第 N指令
執行第 CALL指令
執行 NOP
執行 M指令

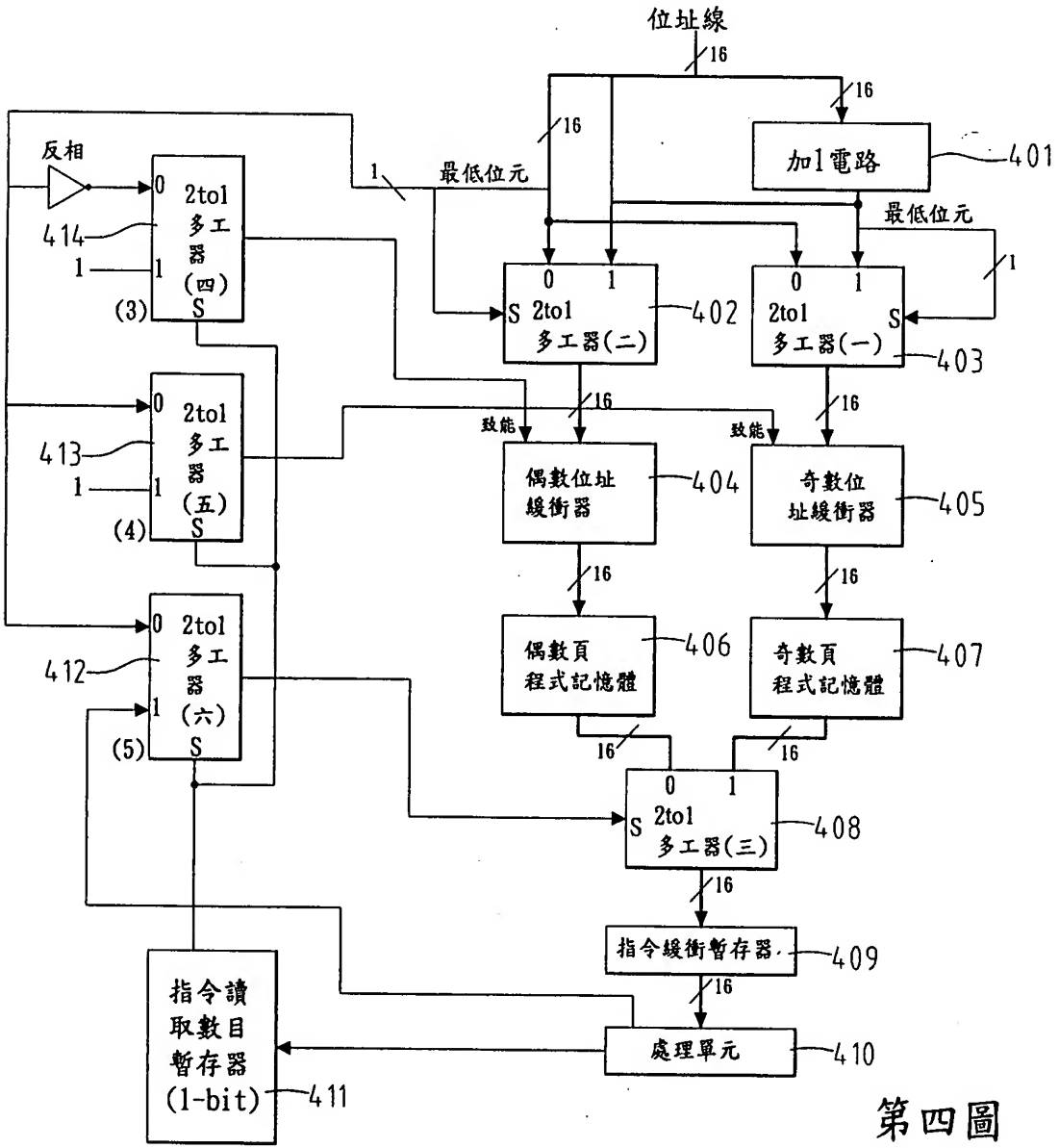
第一圖


T1 T2 T3 T4 T1 T2 T3 T4 T1 T2 T3 T4 T1 T2 T3 T4
讀取第 N+1指令(CALL) N+2指令
讀取第 M指令 M+1指令
讀取第 M+1指令 M+2指令
讀取第 M+2指令 M+3指令
執行第 N指令
執行 NOP
執行第 M指令
執行第 M+1指令

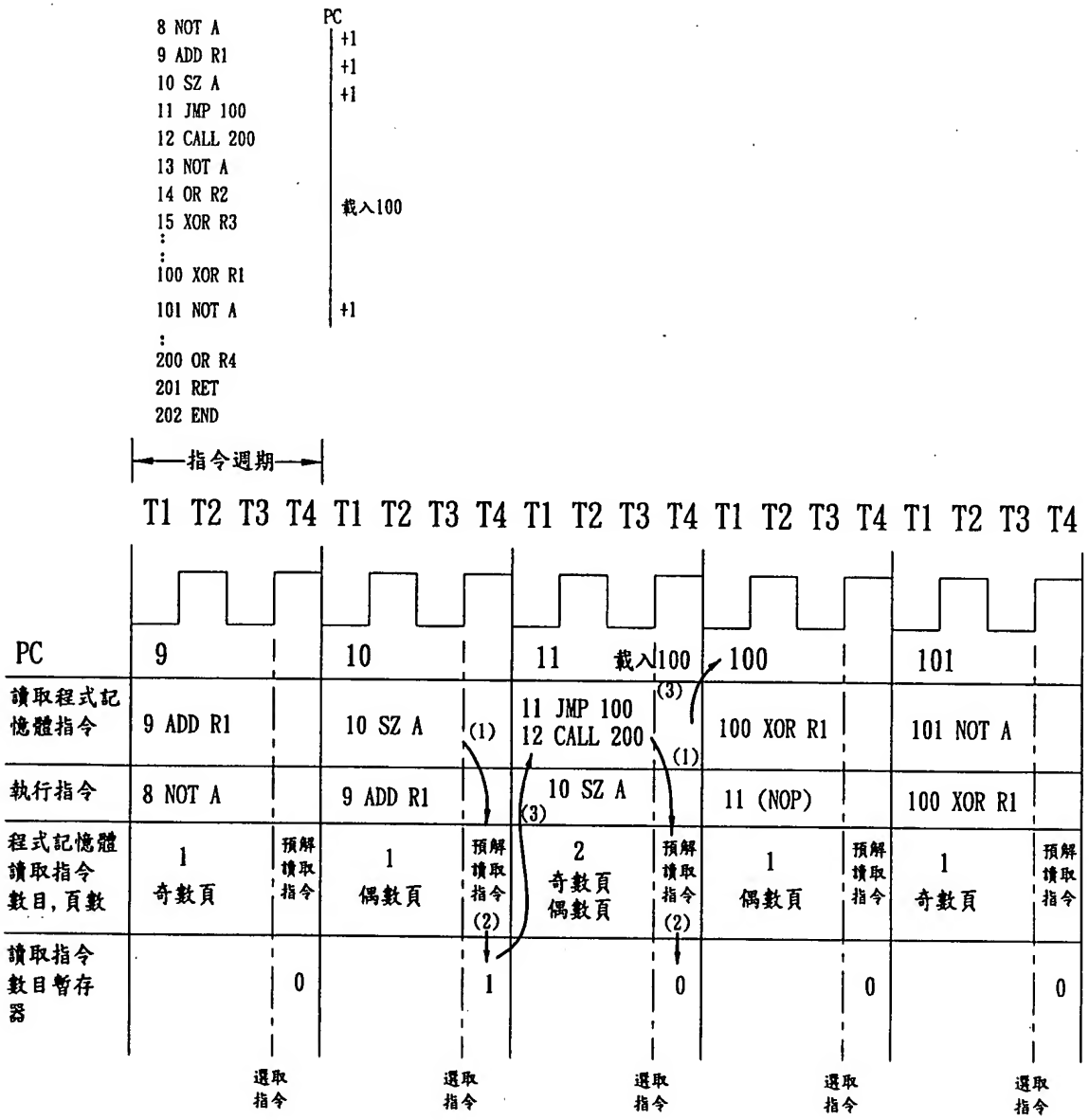
第二圖


301
指令緩衝暫存器
(下一執行指令)
302
預解
讀取
指令
讀取指令為一般
常態指令
303
程式計數器值加1
(PC=PC+1)
307
讀取指令為無條
件跳躍指令
304
程式計數器值
PC=新位址
(新位址於指令上)
308
讀取指令為呼叫
或返回
305
呼叫指令推入堆疊
PC=新位址
(新位址於指令上)
返回指令堆疊放
入PC
PC=新位址
(新位址於堆疊上)
309
讀取指令為條件
式跳躍指令
306
程式計數器值可
能為PC=PC+1
或PC=PC+2
310
設定指令讀
取數目前暫
存器值為0
(只需從程
式記憶體讀
取1指令)
311
設定指令讀
取數目前暫
存器值為1
(需從程式
記憶體讀
取2指令)
312
選取
其
中1
指令
313

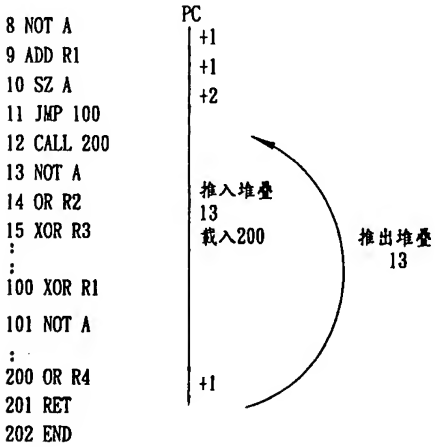
第三圖

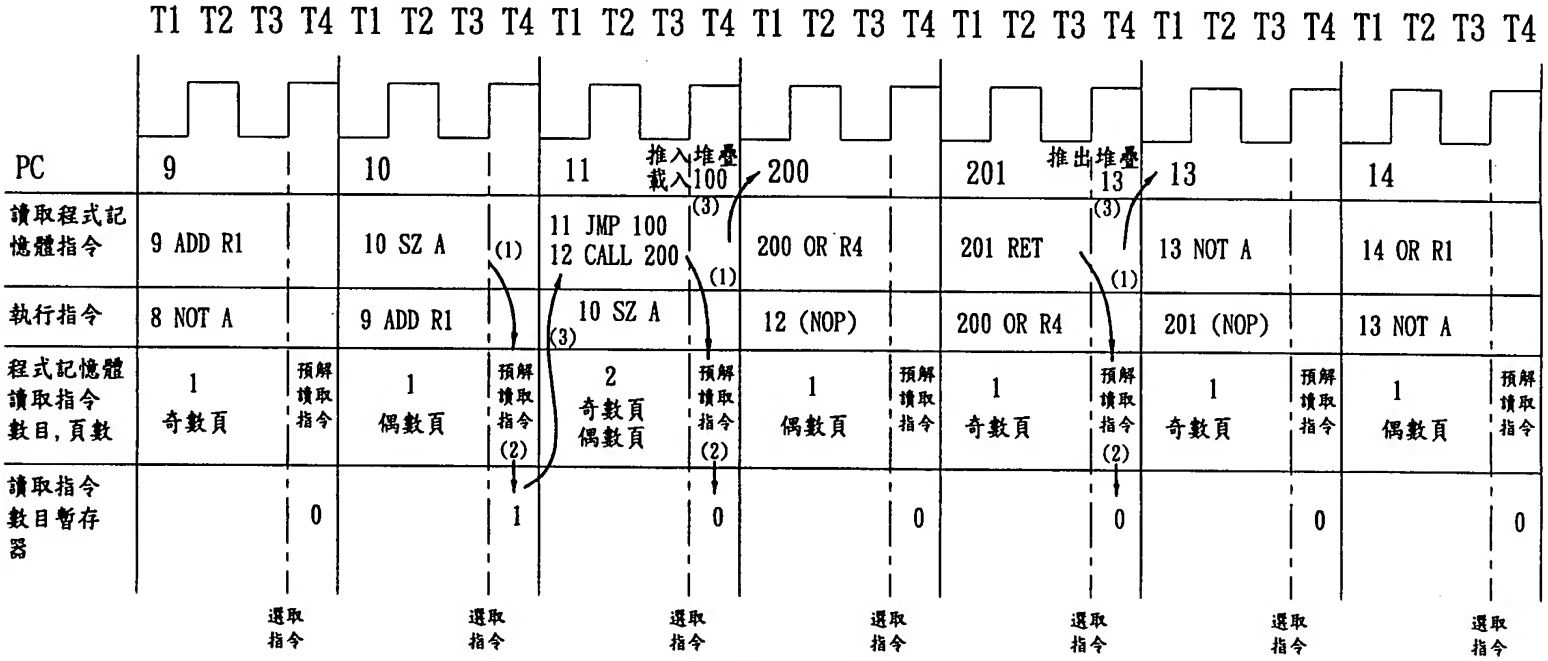

位址線
16
16
16
加1電路
401
最低位元
最低位元
1
反相
414
1
(3)
2to1 多工器 (四)
S
413
1
(4)
2to1 多工器 (五)
S
412
1
(5)
2to1 多工器 (六)
S
0 1
S 2to1 多工器(二)
402
0 1
2to1 多工器(一) S
403
致能
致能
偶數位址緩衝器
404
奇數位址緩衝器
405
偶數頁程式記憶體
406
奇數頁程式記憶體
407
0 1
S 2to1 多工器(三)
408
指令緩衝暫存器
409
處理單元
410
指令讀取數目暫存器 (1-bit)
411

第四圖


8 NOT A
9 ADD R1
10 SZ A
11 JMP 100
12 CALL 200
13 NOT A
14 OR R2
15 XOR R3
100 XOR R1
101 NOT A
200 OR R4
201 RET
202 END
PC
+1
+1
+1
載入100
+1
指令週期
T1 T2 T3 T4 T1 T2 T3 T4 T1 T2 T3 T4 T1 T2 T3 T4 T1 T2 T3 T4
PC
9
10
11
載入100
100
101
讀取程式記憶體指令
9 ADD R1
10 SZ A
11 JMP 100
12 CALL 200
100 XOR R1
101 NOT A
(1)
(3)
(1)
執行指令
8 NOT A
9 ADD R1
10 SZ A
(3)
11 (NOP)
100 XOR R1
程式記憶體讀取指令數目,頁數
1
奇數頁
預解讀取指令
1
偶數頁
預解讀取指令
(2)
2
奇數頁
偶數頁
預解讀取指令
(2)
1
偶數頁
預解讀取指令
1
奇數頁
預解讀取指令
讀取指令數目暫存器
0
1
0
0
0
選取指令
選取指令
選取指令
選取指令
選取指令

第五圖


8 NOT A
9 ADD R1
10 SZ A
11 JMP 100
12 CALL 200
13 NOT A
14 OR R2
15 XOR R3
:
100 XOR R1
101 NOT A
:
200 OR R4
201 RET
202 END
PC
+1
+1
+2
推入堆疊
13
載入200
+1
推出堆疊
13


T1 T2 T3 T4 T1 T2 T3 T4 T1 T2 T3 T4 T1 T2 T3 T4 T1 T2 T3 T4 T1 T2 T3 T4 T1 T2 T3 T4
PC
9
10
11
推入堆疊
載入100
200
201
推出堆疊
13
13
14
讀取程式記憶體指令
9 ADD R1
10 SZ A
(1)
11 JMP 100
12 CALL 200
(3)
(1)
200 OR R4
201 RET
(3)
(1)
13 NOT A
14 OR R1
執行指令
8 NOT A
9 ADD R1
10 SZ A
(3)
12 (NOP)
200 OR R4
201 (NOP)
13 NOT A
程式記憶體讀取指令數目,頁數
1
奇數頁
預解讀取指令
1
偶數頁
預解讀取指令
(2)
2
奇數頁
偶數頁
預解讀取指令
(2)
1
偶數頁
預解讀取指令
1
奇數頁
預解讀取指令
(2)
1
奇數頁
預解讀取指令
1
偶數頁
預解讀取指令
讀取指令數目暫存器
0
1
0
0
0
0
0
選取指令
選取指令
選取指令
選取指令
選取指令
選取指令

第六圖